Perfecto. Aquí tienes **todos los pasos bien detallados, desde cero**, para manejar múltiples claves SSH con GitHub, usando **Git Bash en Windows**, y con un archivo config para mantener todo limpio y profesional. Vamos en **dos iteraciones**:

**🔁 PRIMERA ITERACIÓN — Crear la primera clave SSH, configurar config, añadirla a GitHub**

**🧼 0. Limpieza previa (opcional si ya creaste claves)**

cd ~/.ssh

rm -f id\_ed25519 id\_ed25519.pub config

**🔐 1. Generar una nueva clave SSH para tu cuenta personal de GitHub**

ssh-keygen -t ed25519 -C "tu-correo-personal@gmail.com"

Cuando te pida la ruta del archivo, presiona **Enter** para usar la predeterminada:

/c/Users/PC/.ssh/id\_ed25519

También puedes presionar Enter para dejar la passphrase vacía (o colocar una si deseas más seguridad).

**🧾 2. Crear el archivo de configuración ~/.ssh/config**

nano ~/.ssh/config

Escribe lo siguiente:

# Cuenta personal de GitHub

Host github.com

HostName github.com

User git

IdentityFile ~/.ssh/id\_ed25519

IdentitiesOnly yes

📌 Esto le dice a ssh que use la clave id\_ed25519 cuando se conecte a github.com.

Guarda con Ctrl + O, presiona Enter, y luego Ctrl + X para salir.

**🚀 3. Probar conexión SSH**

ssh -T git@github.com

La primera vez, te dirá:

Are you sure you want to continue connecting (yes/no)?

Escribe:

yes

Y si todo va bien:

Hi BecaLince2024! You've successfully authenticated...

**🌐 4. Copiar la clave pública y subirla a GitHub**

cat ~/.ssh/id\_ed25519.pub

1. Copia toda la línea que empieza con ssh-ed25519.
2. Ve a <https://github.com/settings/ssh/new>
3. Pega la clave ➜ Ponle nombre ("Mi PC personal") ➜ **Add SSH key**

✅ Ya tienes tu primera clave funcionando.

**🔁 SEGUNDA ITERACIÓN — Crear segunda clave para otra cuenta (ej. del trabajo)**

**🔐 1. Generar segunda clave SSH**

ssh-keygen -t ed25519 -C "tu-correo-trabajo@empresa.com"

Cuando pregunte por la ruta del archivo, escribe:

/c/Users/PC/.ssh/id\_ed25519\_work

Presiona Enter varias veces como antes.

**🧾 2. Editar tu archivo ~/.ssh/config para agregar la nueva clave**

nano ~/.ssh/config

Agrega esto al final:

# Cuenta de trabajo de GitHub

Host github-work

HostName github.com

User git

IdentityFile ~/.ssh/id\_ed25519\_work

IdentitiesOnly yes

Guarda y sal con Ctrl + O, Enter, Ctrl + X.

**🌐 3. Subir la nueva clave pública a tu otra cuenta de GitHub**

cat ~/.ssh/id\_ed25519\_work.pub

1. Copia el contenido
2. Entra con tu **cuenta de trabajo de GitHub** ➜ <https://github.com/settings/ssh/new>
3. Pega la clave ➜ Nómbrala ("PC trabajo") ➜ **Add SSH key**

**✅ 4. Usar la cuenta del trabajo con URL personalizada**

Cuando clones repos desde esa cuenta, usa la **URL modificada** con el alias que pusiste en config (github-work):

git clone git@github-work:empresa/repositorio-privado.git

📌 Esto le dice a Git:

"Usa la clave id\_ed25519\_work en lugar de la del host por defecto"

**✅ CONCLUSIÓN: Flujo completo profesional**

1️⃣ Generas claves con nombres distintos

2️⃣ Agregas cada una al archivo ~/.ssh/config

3️⃣ Subes sus claves públicas a cada cuenta de GitHub

4️⃣ Usas el alias del Host en los clones/push (ej. github-work)

¿Te gustaría que te dé ahora un script .sh que automatice esto para una tercera clave o que verifique si todo está bien configurado en tu sistema?